Secure Routing Extensions Crypto API

– SRxCryptoAPI –

Version 0.1.1

User's Manual

March 2015

http://bgpsrx.antd.nist.gov

bgpsrx-dev@nist.gov

Oliver Borchert, Kyehwan Lee, Kotikalapudi Sriram, Doug Montgomery

Index

Introduction

Overview

SRxCryptoAPI is a wrapper that manages Cryptography plugins for BGPSEC path processing. This Software is a wrapper that allows configuring crypto implementations that can be used by software packages such as QuaggaSRx and SRX-Server to switch cryptography implementations. This wrapper allows switching implementations without the need of recompiling QuaggaSRx or SRx-Server. Future versions of both mentioned packages require this API.

This package is shipped with the OpenSSL based implementation for bgpsec path validation and signing called BGPSecOpeSSL. It is possible to write other implementations and link them using the configuration file.

In addition the SRxCryptoAPI software provides a set of tools which allow the generation of keys and certificates used by QuaggaSRx / SRx-Server.

Installation

For Red Hat based systems the software can be installed using the 'yum' installer. The repository information is available on the website http://bgpsrx.antd.nist.gov

For source-based installation read the INSTALL file provided.

SRxCryptoAPI configuration

The API requires a configuration script that is located either in the installation '*/etc/*' directory or in '*./*'. The configuration has the following format:

library_conf = "plugin-config-name"

Specifies the configuration to be used. SRxCryptoAPI allows configuring multiple plugins within the main configuration file. This setting selects the configuration that has to be loaded.

key_volt = "directory for the keys to be stored"

Specifies the location where the public and private keys are stored. This setting can be set programmatically. It is used by the key helper methods that are part of the wrapper API, not the plugin API.

Plugin Configuration Block

Each plugin is configured in a plugin configuration block. Each block is scripted in the following format:

```
plugin-config-name: {
  …
}
```

Within each plugin configuration block is the configuration of the plugin. The library name is required, all other method mappings are needed only in case the plugin uses different function names as proposed by the API itself.

library_name = "custom-library.so"

Specifies the library that will be loaded and wrapped by the SRxCryptoAPI. This parameter is mandatory.

The following configuration settings specify the method mappings in case the plug in uses different method names. In the listing below we map to the default method names. For detailed API function description see the ***srxcryptoapi.h*** header file.

To determine the extend of functionality the plugin provides, consult the documentation provided with the plugin. The minimum requirement a plugin must fulfill are the two methods **validate** and **sign_with_key** where as the method names can differ. If the names differ, a configuration for the method mapping as displayed below must be provided.

method_validate = "validate"

Validate the BGPSec path attribute.

method_sign_with_key = "sign_with_key"

Sign the given path with the private key provided.

Public Key Management Enabled

method_isExtended = "isExtended"

Used to indicate extended functionality of public key management. The return value of this method must be "true" to enable public key management.

method_extValidate = "extValidate"

Allows validation without passing the public keys. Here the plugin provides a key storage where public keys can be held and the plugin determines which stored keys to use. (public key management must be enabled)

method_registerPublicKey = "registerPublicKey"

The plugin provides its own key storage and allows storing the public key provided. (public key management must be enabled)

method_unregisterPublicKey = "unregisterPublicKey"

Remove the previously stored public key from the key storage. (public key management must be enabled)

Private Key Management Enabled

method_isPrivateKeyStorage = " isPrivateKeyStorage"

The plugin allows storing the private key. The return value of this method must be "true" to enable private key management.

method_sign_with_id = "sign_with_id"

Sign the BGPSec path and use the id to specify which previously stored key to use. (private key management must be enabled)

method_registerPrivateKey = "registerPrivateKey"

Register the given private key. (private key management must be enabled)

method_unregisterPrivateKey = "unregisterPrivateKey"

Unregister the previously registered key. (private key management must be enabled)

Key Generation Tools

The tools provided with the SRxCryptoAPI allow to generate private and public keys for test purpose only! Keys should normally be received by a validation cache or through other means.

To setup a key_volt for SRxCryptoAPI perform the following steps:

1. Generate a repository e.g. /var/lib/bgpsec-keys
   This repository will contain the keys and certificates generated by the key generation tools. In this versions the certificates only contain the key.
2. Generate the Keys:
   qsrx-make-key <name>
3. Generate the Certificate
   qsrx_make-cert <name>
4. Publish the cert and install the Key. In addition this step generates a file containing all SKI's for private keys for easy copy and paste into the router configuration.
   qsrx-publish <key>

Support

In case of crashes, please provide a description on how to reproduce the crash and if possible a core dump.

To be informed of bug fixes or ask questions to the community, subscribe to the users email list by sending an email to bgpsrx-users-request@nist.gov with subscribe in the subject.

Questions to the developers and general contact information:

Email: bgpsrx-dev@nist.gov
Web: http://bgpsrx.antd.ist.gov

Developers:

Oliver Borchert oliver.borchert@nist.gov
Kyehwan Lee kyehwanl@nist.gov